

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 10040105 A

(43) Date of publication of application: 13 . 02 . 98

(51) Int. Cl. G06F 9/38

(21) Application number: 09084803

(22) Date of filing: 03 . 04 . 97

(30) Priority: 16 . 04 . 96 US 96 633267

(71) Applicant: INTERNATL BUSINESS MACH  
CORP <IBM>

(72) Inventor:  
KIN S CHAN  
HAN Q RII  
DAN Q NGUIEN

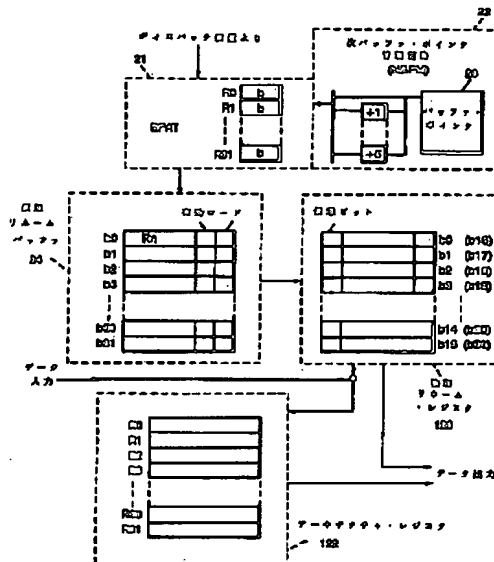
(54) METHOD AND PROCESSOR FOR ALLOCATING  
RENAME REGISTER

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a method and device which make good use of a virtual buffer so as to increase the parallel processing functions for instructions of a pipeline type processor.

SOLUTION: A buffer pointer allocating mechanism is configured and rename buffer are allocated even unless a physical rename register 123 is ready for use during dispatching. According to a buffer pointer allocation table 21, those virtual rename buffers 24 are allocated. When an instruction corresponding to an entry stored in a specific physical rename register 123 is completed, virtual bits set together with respective physical rename registers 123 are inverted, and the results are written in an architecture register 122. Therefore, rename registers can be allocated in dispatching as if more rename buffers than the actual physical rename registers 123 were present.

COPYRIGHT: (C)1998,JPO



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-40105

(43)公開日 平成10年(1998) 2月13日

(51)Int.Cl.<sup>6</sup>

G 0 6 F 9/38

識別記号

3 5 0

庁内整理番号

F I

G 0 6 F 9/38

技術表示箇所

3 5 0 B

審査請求 未請求 請求項の数20 O L (全 15 頁)

(21)出願番号 特願平9-84803

(22)出願日 平成9年(1997) 4月3日

(31)優先権主張番号 0 8 / 6 3 3 2 6 7

(32)優先日 1996年4月16日

(33)優先権主張国 米国 (U S)

(71)出願人 390009531

インターナショナル・ビジネス・マシー  
ンズ・コーポレイション

INTERNATIONAL BUSIN  
ESS MASCHINES CORPO  
RATION

アメリカ合衆国10504、ニューヨーク州  
アーモンク (番地なし)

(72)発明者 キン・エス・チャン

アメリカ合衆国78729、テキサス州、オー  
スチン、パトリッジ・ベンド・ドライブ  
13009

(74)代理人 弁理士 坂口 博 (外1名)

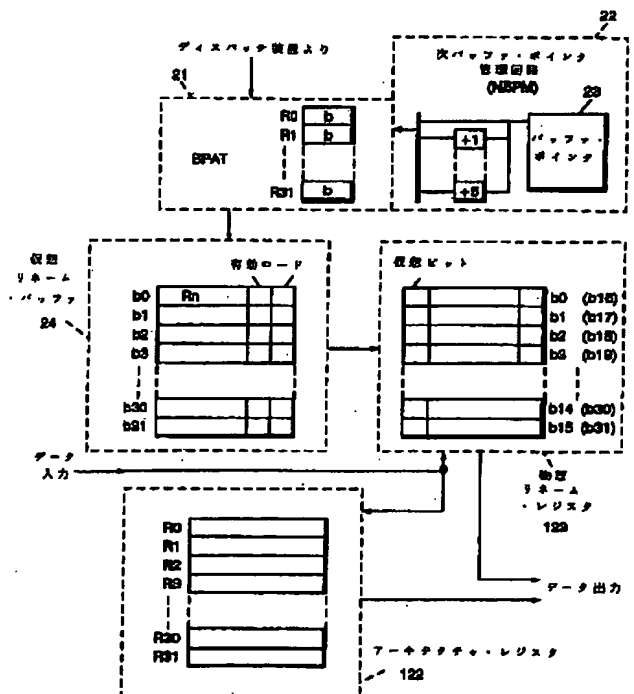
最終頁に続く

(54)【発明の名称】 リネーム・レジスタを割り付ける方法及びプロセッサ

(57)【要約】

【課題】 パイプライン式プロセッサにおける命令の並列処理機能を高めるように仮想バッファを活用した方法及び装置を提供する。

【解決手段】 バッファ・ポインタ割り付け機構を構成して、ディスパッチ中に物理リネーム・レジスタがまだ利用可能になっていなくともリネーム・バッファを割り付ける。バッファ・ポインタ割り付けテーブルにより、これらの仮想リネーム・バッファを割り付ける。特定の物理リネーム・レジスタ内に記憶したエントリに対応する命令が完了したときは、前記各物理リネーム・レジスタと共に設定した仮想ビットを反転させ、その結果をアーキテクチャ・レジスタに書き込む。従って、ディスパッチ時に、実際の物理リネーム・レジスタよりも多くのリネーム・バッファが存在しているかのように、リネーム・レジスタの割り付けを行う。



## 【特許請求の範囲】

【請求項1】一組のN個の物理リネーム・レジスタと、前記N個の物理リネーム・レジスタのうちの一つが命令の結果を受け取り得る状態になる前に、前記N個の物理リネーム・レジスタのうちの一つを該命令に割り付ける事前割り付け回路とを含むプロセッサ。

【請求項2】前記事前割り付け回路はM個の仮想リネーム・バッファを複数の命令に対して割り付ける回路を含み、MはNより大きい請求項1記載のプロセッサ。

【請求項3】前記事前割り付け回路は、前記M個の仮想リネーム・バッファを前記複数の命令に割り付けるバッファ・ポインタ割り付けテーブルと、前記バッファ・ポインタ割り付けテーブルに接続され、巡回形式により逐次前記M個の仮想リネーム・バッファを割り付けるようにこのバッファ・ポインタ割り付けテーブルを制御するバッファ・ポインタ・マネージャとを含む請求項2記載のプロセッサ。

【請求項4】前記N個の物理リネーム・レジスタの各々は、1又は1より多くの仮想ビットを含む請求項2記載のプロセッサ。

【請求項5】前記M個の仮想リネーム・バッファの各々は、前記N個の物理リネーム・レジスタのうちの一に対するポインタを含むように動作可能である請求項4記載のプロセッサ。

【請求項6】2又は2より多くの前記M個の仮想リネーム・バッファはそれぞれ前記N個の物理リネーム・レジスタのうちの同一のものを指示している請求項5記載のプロセッサ。

【請求項7】前記N個の物理リネーム・レジスタのうちの同一のものに関連した仮想ビットは、前記2又は2より多くの前記M個の仮想リネーム・バッファのうちのいずれが、それに割り付けられた命令の結果を書き込み可能であることを表示している請求項6記載のプロセッサ。

【請求項8】更にK個のアーキテクチャ・レジスタを含む請求項2記載のプロセッサ。

【請求項9】 $K \geq M$ である請求項8記載のプロセッサ。

【請求項10】 $N < K \leq M$ である請求項8記載のプロセッサ。

【請求項11】 $K \leq N < M$ である請求項8記載のプロセッサ。

【請求項12】前記K個のアーキテクチャ・レジスタ、前記N個の物理リネーム・レジスタ及び前記M個の仮想リネーム・バッファは1又は1より多くの実行装置に関連される請求項8記載のプロセッサ。

【請求項13】リネーム・レジスタを割り付ける方法において、

第1の命令を受け取るステップであって、前記第1の命令を実行した第1の結果は第1のアーキテクチャ・レジスタに書き込まれるべきものであるステップと、

前記第1のアーキテクチャ・レジスタに対応するように

第1の仮想リネーム・バッファを割り付けるステップとを含み、前記第1の仮想リネーム・バッファは第1の物理リネーム・レジスタに対する第1のポインタを含み、前記割り付けるステップは、前記第1の物理リネーム・レジスタが前記第1の結果を記憶するために利用可能であってなくても実行される、リネーム・レジスタを割り付ける方法。

【請求項14】更に、前記第1の物理リネーム・レジスタが前記第1の結果を記憶するために利用可能であるか否かを判断するステップと、

前記第1の物理リネーム・レジスタが前記第1の結果を記憶するために利用可能であるときに、前記第1の命令を実行するステップと、

前記第1の命令を実行した前記第1の結果を前記第1の物理リネーム・レジスタに記憶するステップを含む請求項13記載の方法。

【請求項15】更に、前記第1の命令を完了させるステップと、前記第1の物理リネーム・レジスタから前記第1のアーキテクチャ・レジスタに前記第1の結果を書き込むステップとを含む請求項14記載の方法。

【請求項16】更に、第2の命令を受け取るステップであって、この第2の命令を実行した第2の結果は第2のアーキテクチャ・レジスタに書き込まれるべきものであるステップと、前記第2のアーキテクチャ・レジスタに対応するように第2の仮想リネーム・バッファを割り付けるステップであって、この第2の仮想リネーム・バッファは前記第1の物理リネーム・レジスタに対する第2のポインタを含むステップとを含む請求項13記載の方法。

【請求項17】更に、前記第1の物理リネーム・レジスタが前記第1のポインタにより指示された第1のアドレスを有するか否かを判断するステップと、

前記第1の物理リネーム・レジスタが前記第1のポインタにより指示された前記第1のアドレスを有するときは、前記第1の命令を実行するステップと、

前記第1の命令を実行した前記第1の結果を前記第1の物理リネーム・レジスタに記憶するステップと、

前記第1の命令を完了させるステップと、前記第1の物理リネーム・レジスタから前記第1のアーキテクチャ・レジスタに前記第1の結果を書き込みステップと、

前記第1のアドレスを前記第2のポインタにより指示された第2のアドレスに変換するステップと、

前記第2の命令を実行するステップと、前記第1の物理リネーム・レジスタに前記第2の命令を実行した前記第2の結果を記憶するステップと、

前記第2の命令の完了させるステップと、

前記第2の物理リネーム・レジスタから前記第2のアーキテクチャ・レジスタに書き込むステップとを含む請求項16記載の方法。

【請求項18】前記第1のアーキテクチャ・レジスタは、プロセッサにおける実行装置に関連したK個のアーキテクチャ・レジスタのうちの一つであり、かつ前記第1の物理リネーム・レジスタは前記実行装置に関連されたN個の物理リネーム・レジスタのうちの一つであり、かつ前記第1の仮想リネーム・バッファは前記実行装置に関連したM個の仮想リネーム・バッファのうちの一つであり、かつ $N < M$ である請求項13記載の方法。

【請求項19】前記完了するステップは、(1)前記第1の命令、及び前記第1の命令に先行する全ての命令が実行されたか、(2)割り込みが発生したか、及び(3)誤った分岐をしたかについて判断するステップを含む請求項15記載の方法。

【請求項20】プロセッサにおいて、実行装置と、前記実行装置に関連されたK個のアーキテクチャ・レジスタと、前記実行装置に関連されたN個の物理リネーム・レジスタと、M個の仮想リネーム・バッファ(ただし、 $N < M$ )と、第1の命令を受け取るディスパッチ装置であって、前記第1の命令を実行した第1の結果は前記K個のアーキテクチャ・レジスタのうち第1のものに書き込まれるべきものである前記ディスパッチ装置と、前記K個のアーキテクチャ・レジスタのうち前記第1のものに対応するように前記M個の仮想リネーム・バッファのうち第1のものを割り付けるバッファ・ポインタ割り付けテーブルであって、前記M個の仮想リネーム・バッファのうち前記第1のものは前記N個の物理リネーム・レジスタのうち第1のものに対する第1のポインタを含む前記バッファ・ポインタ割り付けテーブルと、前記N個の物理リネーム・レジスタの前記第1のものが前記第1のポインタに対応する第1のアドレスを含むか否かを判断する回路とを含む、前記実行装置は、前記N個の物理リネーム・レジスタのうち前記第1のものが前記第1のポインタに対応する前記第1のアドレスを含むときに、前記第1の命令を実行するものであって、更に、前記第1の命令を実行した前記第1の結果を前記N個の物理リネーム・レジスタのうち前記第1のものに記憶する回路と、前記第1の命令を完了させる回路と、前記N個の物理リネーム・レジスタの前記第1のものから前記K個のアーキテクチャ・レジスタのうち前記第1のものに前記第1の結果を書き込む回路とを含む、前記ディスパッチ装置は第2の命令を受け取り、前記第

2の命令を実行した第2の結果は前記K個のアーキテクチャ・レジスタのうち第2のものに書き込まれるべきものであって、

前記バッファ・ポインタ割り付けテーブルは、前記K個のアーキテクチャ・レジスタのうち前記第2のものに対応するように、前記M個の仮想リネーム・バッファのうち第2のものを割り付け、前記M個の仮想リネーム・バッファのうち前記第2のものが前記N個の物理リネーム・レジスタのうち前記第1のものに対する第2のポインタを含むものであって、更に、

前記第1のアドレスを前記第2のポインタにより指示された第2のアドレスに変換する回路を含み、

前記実行装置は前記第2の命令を実行するものであって、更に、

前記第2の命令を実行した前記第2の結果を前記N個の物理リネーム・レジスタのうち前記第1のものに記憶する回路を含むプロセッサ。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】本発明は、データ処理システム及びデータ処理方法に関し、特にプロセッサ内で命令のデスパッチ中に利用可能なリネーム・バッファ数を増加させ、かつデスパッチ帯域幅を増加させる装置及び方法に関する。

【0002】

【従来の技術】一般的に、コンピュータは、フォン・ノイマン・アーキテクチャに従って設計されるが、このアーキテクチャはハンガリー生まれの数学者ジョン・フォン・ノイマンの研究によるとされた、マイクロコンピュータを含むごく一般的に用いられるコンピュータの設計特徴に対する一解決方法である。フォン・ノイマン・アーキテクチャは、ストアド・プログラム、即ちコンピュータに永久的に記憶でき、かつ、符号化される形式のために、マシン・ベースの命令により操作又は自ら変更することができるという概念と同意語である。よく知られた逐次的な処理の概念、即ち一時に一命令の複数オペレーションに対する解決方法は、フォン・ノイマン・アーキテクチャの特徴である。

【0003】このような逐次命令のコンピュータ・アーキテクチャが有する問題は、論理回路が如何に早く実行できるかによってその処理速度が制限されてしまうことである。この問題に対する一つの解決方法は、設計及びスーパスカラ・マイクロプロセッサ・アーキテクチャの設計及び使用にあった。これは、マイクロプロセッサに1クロック・サイクル当り多数の命令を実行できるようにするものである。このようなスーパスカラ・プロセッサには、IBMにより作成されたPowerPCプロセッサがある。

【0004】このようなスーパスカラ・プロセッサでは、命令装置がプロセッサ内の種々の実行装置に対して

一度に複数の命令をディスパッチする。しかし、このようなアーキテクチャでは、数個の命令がプログラム内の他の命令の完了に依存しているため、別の問題が発生する。換言すれば、2つのオペランドを加算するオペレーションは、他の命令が完了して、加算するオペランドのうちの一つが発生するまで、待機しなければならないことがある。

【0005】更に、このようなプロセッサは、可能分岐条件を有する命令を読み出す分岐処理装置としばしば呼ばれるものを使用するものであり、これによって前の命令の結果に従ってプログラム命令フロー内で2つの異なるパスを取ることができる。このような分岐処理装置は、プロセッサがどのプログラム分岐を可能性として取るのかを予測して進行し、その分岐内の後続命令を備えて実行し始める。このプロセッサ内の完了装置は、ディスパッチから実行まで命令をトラッキングする機構を提供して、これらの命令をプログラム順に「完了」させる。命令の完了は、その命令実行の結果をアーキテクチャ・レジスタへコミットすることを含む。プログラム順の完了によって正確なアーキテクチャ上の状態を保証し、プロセッサは、予測を誤った分岐や他の介入や割り込みから回復しなければならない。「完了した」命令の結果は、アーキテクチャ・レジスタに書き込まれる。

【0006】プロセッサは、非プログラム順序の実行の過程で与えられたレジスタ・ファイルの位置に対する競合を避けるために、完了装置によりアーキテクチャ・レジスタにコミットされるのに先立って、命令の結果を記憶するリネーム・レジスタを設けてもよい。種々の実行装置及びプロセッサにおいてこれらに関連するアーキテクチャ・レジスタのそれぞれについて、数個のリネーム・レジスタ又はバッファが設けられてもよい。

【0007】ディスパッチ装置がその実行装置に命令をディスパッチする際は、その命令の結果用にリネーム・レジスタが割り付けられる。命令がデータ依存性のために実行装置に関連された予約ステーションにディスパッチされるのであれば、ディスパッチャーは更に実行装置にタグを与えて、命令の完了によりどのリネーム・レジスタが要求データを転送するのかを識別させる。リネーム・レジスタからデータが得られるときは、ペンディング実行を開始することができる。

【0008】命令が例外なしに完了キューから撤収され、かつ、完了キューの中のこれに先行する推論的な分岐条件が正しく決定された後に、完了装置によって命令の結果はリネーム・レジスタからアーキテクチャ・レジスタに転送される。推論的な分岐条件を誤って予測したことが判ったときは、その分岐に続く推論的に実行した命令が完了キューから消去され、かつこれらの命令の結果もリネーム・レジスタから消去される。

【0009】命令をディスパッチする際の隘路は、ディスパッチ装置が全てのリネーム・レジスタを割り付けた

ときに発生することがある。これはリネーム・レジスタが空きとなり割り付け可能になるまで、ディスパッチ装置を停止させる要因となる。

【0010】従来技術による処理手段では、以上の問題を除去するために複雑な制御及びデータ・フローを利用して又は内容参照可能メモリを使用して、リネーム機構を実施していた。加えて、増大する数の潜在的な推論命令をサポートするために、更に多くの実行装置を実施すれば、更に多くのリネーム・レジスタを必要とする。

10 【0011】従って、当該技術分野では、複数の命令をディスパッチする際に、前述の非効率さをなくすプロセッサ・アーキテクチャに対する技術が要求されている。

【0012】

20 【発明が解決しようとする課題】従って、本発明の目的は、内容参照可能メモリ及び複合リネーム・レジスタ制御に対する必要性をなくすことである。また本発明の目的は、物理的に実施するよりも多くのリネーム・レジスタ（仮想リネーム・レジスタ）を割り付けることができる仮想リネーム機構を提供することである。更に本発明の目的は、プロセッサの設計を大幅に変更することなく、リネーム・レジスタの数を変更することができる処理手段を提供することである。

【0013】

30 【課題を解決するための手段】本発明は、命令をディスパッチ装置から実行及び完了装置へディスパッチする際の隘路を除くことにより、以上の要求を満足させる。これは、ディスパッチ装置に存在する物理リネーム・バッファよりも多くのリネーム・バッファを割り付けられるようにして達成される。これはディスパッチ装置が実行装置に複数の命令を連続してディスパッチできるようにさせる。各実行装置が実行すべき命令を受け取る際は、ディスパッチ装置によって割り付けられた仮想リネーム・バッファの「アドレス」に一致する利用可能な物理リネーム・レジスタが存在するか否かをチェックする。もしこれが存在するのであれば、実行装置は命令を実行してその命令の結果を物理リネーム・レジスタに書き込む。

40 【0014】命令が完了すると、その結果が物理リネーム・レジスタからアーキテクチャ・レジスタに書き込まれ、これによって物理リネーム・レジスタが開放される。

【0015】以上の処理は、各物理リネーム・バッファに付加された仮想ビット・アドレスを使用して実施される。

【0016】以上、以下に続く本発明の詳細な説明をよく理解できるように、本発明の特徴及び技術的な効果を概要的に説明した。以下、本発明の請求の要旨をなす本発明の更なる特徴及び効果を説明する。

【0017】

50 【発明の実施の形態】以下の説明では、本発明の十分な

理解が得られるように、特殊なワード又はバイト長など特定の詳細説明を多数行う。しかし、当該技術分野に習熟する者において、特定の詳細説明がなくとも本発明を実施できることは明らかである。他の例において、周知の回路は、必要以上の詳細な説明により本発明を不明確にしないように、ブロック形式により示されている。大抵の部分において、タイミング条件等に関する詳細な説明については、それが本発明を完全に理解するために必要であるというのでない限り、省略した。

【0018】ここで図面を参照する。ただし、説明する各要素は必ずしも図示のスケール通りではなく、また同一又は同様の要素はいくつかの図を通して同一の参照番号により識別される。

【0019】以上で説明したように、レジスタのリネーム技術はマイクロプロセッサ設計において周知である。リネームした（一時）レジスタをアーキテクチャ機構に割り付けることによって、推論結果はアーキテクチャ的に明確な順序を保証できるまでアーキテクチャ機構にコミットされないのが、命令の推論実行は順序に無関係に進めることができる。

【0020】図1を参照すると、システム・バス712にバス103及び104を介して接続されたプロセッサ100が示されており、このシステム・バス712はそれぞれアドレス・バス101及びデータ・バス102を含む。プロセッサ100は、公知の中央処理装置（例えば、IBMにより製作されたPowerPCプロセッサ）であってもよく、また図1に示す回路のいくつか又は全てを有する。

【0021】バス103及び104はバス・インターフェイス装置（BIU）105に接続されている。命令キャッシュ（Iキャッシュ）107はBIU105及び命令装置108の逐次フェッチャー109に接続されている。命令キャッシュ107は、複数のタグ・ビットを有するものが可能であって、命令記憶管理装置（IMMU）121を介して命令装置108にも接続されている。

【0022】命令装置108には、逐次フェッチャー109、分岐処理装置（BPU）110、命令キュー111及びディスパッチ装置112が含まれている。

【0023】典型的な実施では、BIU105がデータ・バス102からデータを受け取って、データ・キャッシュ（Dキャッシュ）106を介してロード／ストア装置115に転送する。命令キャッシュ107はBIU105から命令を受け取って、これらの命令を逐次フェッチャー109に転送する。

【0024】BPU110は、複数の分岐命令を受け取り、複数の条件分岐に基づきルックアヘッド・オペレーションを実行してこれらを速やかに解決する動作が可能である。命令キュー111は逐次フェッチャー109から複数の命令を受け取って、これらをディスパッチ装置

112に転送する。ディスパッチ装置112は、ロード／ストア装置115、浮動小数点装置116、整数ユニット114及びシステム・レジスタ装置113のようないくつかの実行装置のうちのいずれか一つに命令をディスパッチするように動作可能である。これらの実行装置は更に完了装置117にも接続されており、完了装置117は、複数の命令を実行を介してディスパッチからトラッキングし、次いでプログラム順序により撤収させる即ち「完了」させる。完了装置117は完了バッファのキューを備えている。完了装置117は更に逐次フェッチャー109にも接続されている。

【0025】更に、浮動小数点装置116には、浮動小数点（FP）リネーム・レジスタ125と共に浮動小数点レジスタ（FPR）ファイル124が接続されている。

【0026】整数ユニット114には、汎用レジスタ（GPR）ファイル122及び関連する汎用（GP）リネーム・レジスタ123が接続されている。

【0027】実行装置のうちのいずれか一つが本発明のアーキテクチャを備えることができる。しかし、図1では、複数の仮想リネーム・バッファ24に接続されたバッファ・ポインタ・アサインメント・テーブル（BPAT）21と、関連の次バッファ・ポインタ管理（NBPM）回路22とが示されており、これら仮想リネーム・バッファ24はGPRファイル122と、GPリネーム・レジスタ123とに関連されている。BPAT21は複数のバッファ・ポインタを備えている。BPAT21は、アーキテクチャ・ポインタをバッファ（物理）ポインタにマップさせるために用いられる。複数の仮想リネーム・バッファ24は複数のアーキテクチャ・ポインタを備えている。複数の物理リネーム・レジスタ123は、データ（複数のオペランド）を備えており、このデータは複数命令を実行するために用いられ、かつ分岐又は割り込みを原因として取り消される対象である。複数のアーキテクチャ・レジスタ122は各アーキテクチャ・レジスタに対応するデータ（複数のオペランド）を備えている。

【0028】次に図2を参照すると、本発明の実施例がブロック図の形式により示されている。BPAT21は各アーキテクチャ・レジスタ・ポインタRK（ただし、Kは正の整数である。）に対するエン트리bにより定義される。ディスパッチ時に、各目標レジスタに対して利用可能な一つのバッファを割り付けることにより、複数の目標レジスタが複数のリネーム・バッファに割り付けられる。これらのリネーム・バッファが割り付けられ、アーキテクチャ・ソース・レジスタは、BPATテーブルにおいてルックアップされる。その結果、複数のソース・リネーム・ポインタはそのテーブル中のアーキテクチャ・レジスタにより指示されたエン트리から直接ルックアップされる。アーキテクチャ・スロットにおけるリ

ネーム・ポインタが無効とマークされている場合、そのオペランドはアーキテクチャ・レジスタから利用可能である。そうでないときは、BPATテーブルにおけるポインタにより指示されたリネーム・バッファに従ってオペランドをマークする。バッファの割り付けは、第1のエントリに戻るように、循環的に行われてもよい。これらのバッファは、物理バッファよりも多くの仮想バッファが存在するので、仮想識別により割り付けられる。命令に対するバッファ割り付けは、命令と共に実行要素に送出される。命令が完了する際は、関連するバッファ・レジスタはアーキテクチャ・レジスタに書き込まれて、そのバッファが割り付け用に利用可能にされる。

【0029】従って、BPAT 21はディスパッチ装置112からディスパッチされた各命令の結果用に仮想リネーム・バッファ24を割り付ける。次バッファ・ポインタ管理(NBPM)回路22内のロジックは、バッファ・ポインタ・レジスタ23を用いており、新しい各命令用に仮想リネーム・バッファ24を割り付けたためのアルゴリズムを利用する。前述のように、このアルゴリズムは逐次的な循環により行われるものでもよい。しかし、本発明は、非循環又は非逐次的な割り付け/取り消しアルゴリズムのように、もう少し複雑なバッファ管理機構によりサポートが可能であり、かつ本発明の範囲内にある。

【0030】仮想リネーム・バッファ24は各アーキテクチャ・レジスタ122RKに対応するバッファbMを有することができる。図2に示す例では、32個の仮想リネーム・バッファ24(M=32)及び32個のアーキテクチャ・レジスタ122(K=32)が存在する。しかし、存在する物理リネーム・レジスタ123(N=16)は16個のみである。

【0031】1個の仮想リネーム・バッファ24につき1個の有効ビット(V)が存在する。リネーム・バッファ24が割り付けられると、Vビットがセットされる。命令が完了又は取り消しされると、Vビットがリセットされる。Vビットがリセットされると、対応するリネーム・レジスタ123は割り付けられず、現在データはアーキテクチャ・レジスタ122内にある。

【0032】更に、1個の仮想リネーム・バッファ24につき1個のロード・ビット(L)が存在する。Lビットは、オペランドがリネーム・バッファ・エントリ24に書き込まれると、指定される。命令が完了又は取り消されると、否定にされる。Vビットがセットされると、Lビットは、リネーム・バッファ24が有効データにより書き込まれたことを表示する。

【0033】各物理リネーム・レジスタ123は更に仮想ビットも有する。この仮想ビットは、ディスパッチ装置112によってまだ利用可能になっていないバッファを予め割り付けられるようにする。この例では、32個のアーキテクチャ・レジスタ122と共に16個の物理

リネーム・レジスタ123が存在する。ディスパッチ時に、リネーム・レジスタ123は、32個のリネーム・レジスタが存在するかのように割り付けられる。しかし、仮想ビットの状態は、いずれのリネーム・バッファ24が物理リネーム・レジスタ123に割り付けられているのかを表す。従って、物理リネーム・レジスタb0は命令に仮想リネーム・バッファb0として割り付けられ、その仮想ビットは0(例えば否定)にマークするようにしてもよい。続いて、物理リネーム・レジスタb0を仮想リネーム・バッファb16として他の命令に割り付けられてもよい。目標としての物理リネーム・レジスタb0を用いる命令が完了すると、仮想ビットが1に反転され即ちセットされ、かつ第2の命令がそのリネーム目標としてこの物理リネーム・レジスタを使用するのが許可される。要するに、物理リネーム・バッファの結果がアーキテクチャ・レジスタにコミットされるときは、物理リネーム・バッファ用の仮想ビットがトグルされる。従って、この物理リネーム・バッファを用いる次の命令は、進行して実行することができる。

【0034】この例において、各仮想リネーム・バッファ24は、16個の物理リネーム・レジスタ123のうちの一つを指示するように、5ビットのポインタを備えることができる。5ビットのポインタ内の最上位ビットが0であれば、そのポインタは物理リネーム・レジスタb0...b15のうちの一つを指示することができる。仮想リネーム・バッファ内のポインタにおける最上位ビットがセットされると、物理リネーム・レジスタb16...b31のうちの一を指示することができる。

【0035】仮想ビットのための他の構成は、物理リネーム・レジスタ123の数が仮想リネーム・バッファ24の数の1/2より少ない場合に、1より多い仮想ビットを有することを含め、本発明の範囲内にあることが明らかである。更に、この構成において、本発明の構成においてアーキテクチャ・レジスタ122の数とリネーム・レジスタ123の数との間には何の関係もないので、アーキテクチャ・レジスタ122より多くの物理リネーム・レジスタ123を実施することができる。

【0036】逐次循環形式により仮想リネーム・バッファを割り付けることにより、推論命令の取り消しが容易にサポートされる。

【0037】本発明の効果のうちの一つは、ディスパッチ処理中にディスパッチ装置112において製作された停止条件をなくすと共に、この停止条件を実行フェーズに配置することである。従来技術の構成では、ディスパッチ装置112が特定の実行装置に関連した全ての物理リネーム・レジスタを割り付けたときは、この特定の実行装置にもはや命令をディスパッチすることはできないので、ディスパッチ装置112に「隘路」が形成されていた。本発明は、物理的に存在するリネーム・レジスタよりも多くのリネーム・レジスタを割り付けられるよう

にして、隘路を処理の実行部へ下方移動させることにより、このような隘路を解消している。換言すれば、特定の実行装置は、特定の命令に割り付けられた仮想リネーム・バッファに対応する物理リネーム・レジスタがまだ利用可能にならないときに、停止することができる。従って、ここで、停止条件は独立した各実行装置内の命令の実行に依存するものであって、ディスパッチ装置112に依存するものではない。

【0038】図3を参照すると、命令がプロセッサ100内で進行する典型的な処理が示されている。まず、ステップ31において命令がディスパッチされ、次にステップ32において実行フェーズに進行し、最後にステップ33において完了段階に進む。

【0039】次に図4を参照すると、ディスパッチのステップ31の更に詳細なフローチャートが示されている。ステップ41においてディスパッチが開始されてステップ42に進み、ディスパッチ装置112から実行装置に命令をディスパッチする。次に、ステップ43において、BPAT21はディスパッチされた命令の結果により次に利用可能な仮想リネーム・バッファ24を割り付ける。ステップ44において、NBPM回路22は次の仮想リネーム・バッファ24に進む。ステップ45において、処理がステップ42に戻されてBPAT21により割り付けられる仮想リネーム・バッファ24が残っているときは、他の命令をディスパッチさせる。それ以外は、仮想リネーム・バッファ24が利用可能になるまで、処理が停止する。

【0040】ディスパッチ装置112からの命令のディスパッチは物理リネーム・レジスタ123の数に依存することなく、実施される仮想リネーム・バッファ24の数に依存するだけである。その数は図2に示すこの例においてアーキテクチャ・レジスタ122の数と同一である。

【0041】次に図5を参照すると、ステップ32における実行フェーズの更なる詳細が示されている。ステップ51において、実行装置は、命令を割り付けられた仮想リネーム・バッファ24と共に受け取る。

【0042】次に、ステップ52において、割り付けた仮想リネーム・バッファ24により指示されている目標の物理リネーム・レジスタ123がまだ利用可能になっていないときは、停止条件を起動させてもよい。これは、物理リネーム・レジスタの仮想ビットがセットされている、又はその逆である間に、その仮想リネーム・バッファの最上位ビットが否定にされているときに、発生する。例えば、その命令を仮想リネーム・バッファb0に割り付けてもよい。仮想リネーム・バッファb0内のポインタは00000となり、物理リネーム・レジスタb0を指示する。ステップ52において、物理リネーム・レジスタb0がセットにされた又は否定にされた仮想ビットを有するの否かを判断する。仮想ビットがセッ

トされているのであれば、仮想リネーム・バッファb16から他の結果を受け取るためにその物理リネーム・レジスタが予約される。他の命令が完了しているときにのみ、仮想ビットが反転されて（この例では否定にされて）、ステップ51において受け取った命令をステップ53において実行できるようにさせ、その命令の実行の結果が物理リネーム・レジスタb0に記憶される（ステップ54）。

【0043】次に図6を参照すると、完了のステップ33が更に詳細に示されている。ステップ61において、この命令及びこの命令の前にディスパッチされた全ての命令が実行されたか否かについてチェックされる。この命令及び前にディスパッチされた全ての命令を実行した、かつ割り込みを発生させていない、かつ解決されるべき分岐が間違っていないときは、処理をステップ62に進める。ステップ62において、物理リネーム・レジスタ123に記憶されたその命令の結果をアーキテクチャ・レジスタ122に書き込む。従って、前述の例において、ステップ51において受け取り、かつステップ54において物理リネーム・レジスタb0に記憶した命令の結果は、アーキテクチャ・レジスタR0がBPAT21における仮想リネーム・バッファb0に割り付けられたのであれば、アーキテクチャ・レジスタR0に書き込まれてもよい。その後、ステップ63において、物理リネーム・レジスタb0に対応する仮想ビットは反転される。

【0044】以下は命令のディスパッチ及び仮想リネーム・バッファ24の割り付けに関する他の例である。命令I0及びI1はディスパッチ装置112により以下のようにディスパッチされてもよい。

I0 add R1, R2, R3

I1 add R3, R1, R2

【0045】命令I0はアーキテクチャ・レジスタR2及びR3内のオペランドを加算し、かつその結果をアーキテクチャ・レジスタR1に書き込むことである。命令I1はアーキテクチャ・レジスタR2における値と、アーキテクチャ・レジスタR1に記憶した（命令I0の結果として発生した）値とを加算して、その結果をアーキテクチャ・レジスタR3に書き込む。

【0046】BPAT21はアーキテクチャ・レジスタR1用の仮想リネーム・バッファb0を割り付けることもできる。次いで、命令I0及びI1は整数装置114のような実行装置のうちの一つに送出される。その後、命令I2がディスパッチ装置112によりディスパッチされてもよい。

I2 R6, R4, R5

【0047】命令I2はR4及びR5の値を加算し、かつこれらをR6に記憶する。BPAT21はR6を仮想リネーム・バッファb16に割り付けてもよい。

【0048】命令I0に割り付けられた仮想リネーム・

10

20

30

40

50



バッファb0に対応するポインタは、00000（物理リネーム・レジスタb0を指示する）であり、一方命令I2に割り付けられた仮想リネーム・バッファb16に対応するポインタは10000（物理リネーム・レジスタb0を指示する）である。仮想リネーム・バッファb0及びb16の両者は物理リネーム・レジスタ123内の物理リネーム・レジスタb0を指示することに注意すべきである。しかし、ディスパッチ装置112からのこれら命令I0及びI2のディスパッチは、停止されることなく、継続するようにされ、次いでこれらの命令は整数装置114に転送される。整数装置114は命令I0を実行し、かつ物理リネーム・レジスタb0に対応する仮想ビットが否定されているのであれば、その結果は物理リネーム・レジスタb0に書き込まれる。命令I2は、完了装置117により命令I0が完了され、物理リネーム・レジスタb0に記憶されたその結果がアーキテクチャ・レジスタR0に書き込まれ、物理リネーム・レジスタb0が解放されるまで、整数装置114によって処理されない。これを実行すると、物理リネーム・レジスタb0に対応する仮想ビットは、反転されるので、セットされ、従って命令I2が実行可能になる。これは、その結果が物理リネーム・レジスタ（16進16）（b16）に書き込まれることになるからである。

【0049】本発明を実施する代表的なハードウェア環境は、発明の要旨に従ってワークステーション713の典型的なハードウェア構成を示す図7に記載されている。このワークステーション713はプロセッサ（CPU）100（図1を参照）、及びシステム・バス712を介して相互接続された多数の他の装置を有する。図7に示すワークステーション713は、ランダム・アクセス・メモリ（RAM）714と、読み出し専用メモリ（ROM）716と、システム・バス712にディスク装置720及びテープ・ドライブ740のような周辺装置を接続する入出力（I/O）アダプタ718と、キー・ボード724、マウス726及び／又はシステム・バス712にタッチ・スクリーン装置（図示なし）のようなユーザ・インターフェイス装置を接続するインターフェイス・アダプタ722と、ワークステーション713をデータ処理ネットワークに接続する通信アダプタ734と、システム・バス712をディスプレイ装置738に接続するディスプレイ・アダプタ736とを含む。プロセッサ100は単一の集積回路上に存在していてもよい。

【0050】本発明及びその効果を詳細に説明したが、請求の範囲により定義されているように本発明の精神及び範囲から逸脱することなく、ここで種々の変更、置換及び好意を行うことができることを理解すべきである。

【0051】まとめとして、本発明の構成に関して以下の項を開示する。

【0052】（1） 一組のN個の物理リネーム・レジ

スタと、前記N個の物理リネーム・レジスタのうちの一つが命令の結果を受け取り得る状態になる前に、前記N個の物理リネーム・レジスタのうちの一つを該命令に割り付ける事前割り付け回路とを含むプロセッサ。

（2） 前記事前割り付け回路はM個の仮想リネーム・バッファを複数の命令に対して割り付ける回路を含み、MはNより大きい（1）記載のプロセッサ。

（3） 前記事前割り付け回路は、前記M個の仮想リネーム・バッファを前記複数の命令に割り付けるバッファ・ポインタ割り付けテーブルと、前記バッファ・ポインタ割り付けテーブルに接続され、巡回形式により逐次前記M個の仮想リネーム・バッファを割り付けるようにこのバッファ・ポインタ割り付けテーブルを制御するバッファ・ポインタ・マネージャとを含む（2）記載のプロセッサ。

（4） 前記N個の物理リネーム・レジスタの各々は、1又は1より多くの仮想ビットを含む（2）記載のプロセッサ。

（5） 前記M個の仮想リネーム・バッファの各々は、前記N個の物理リネーム・レジスタのうちの一に対するポインタを含むように動作可能である（4）記載のプロセッサ。

（6） 2又は2より多くの前記M個の仮想リネーム・バッファはそれぞれ前記N個の物理リネーム・レジスタのうちの同一のものを指示している（5）記載のプロセッサ。

（7） 前記N個の物理リネーム・レジスタのうちの同一のものに関連した仮想ビットは、前記2又は2より多くの前記M個の仮想リネーム・バッファのうちのいずれが、それに割り付けられた命令の結果を書き込み可能であるかを表示している（6）記載のプロセッサ。

（8） 更にK個のアーキテクチャ・レジスタを含む（2）記載のプロセッサ。

（9）  $K \geq M$ である（8）記載のプロセッサ。

（10）  $N < K \leq M$ である（8）記載のプロセッサ。

（11）  $K \leq N < M$ である（8）記載のプロセッサ。

（12） 前記K個のアーキテクチャ・レジスタ、前記N個の物理リネーム・レジスタ及び前記M個の仮想リネーム・バッファは1又は1より多くの実行装置に関連される（8）記載のプロセッサ。

（13） リネーム・レジスタを割り付ける方法において、第1の命令を受け取るステップであって、前記第1の命令を実行した第1の結果は第1のアーキテクチャ・レジスタに書き込まれるべきものであるステップと、前記第1のアーキテクチャ・レジスタに対応するように第1の仮想リネーム・バッファを割り付けるステップとを含み、前記第1の仮想リネーム・バッファは第1の物理リネーム・レジスタに対する第1のポインタを含み、前記割り付けるステップは、前記第1の物理リネーム・レジスタが前記第1の結果を記憶するために利用可能であ

ってもなくても実行される、リネーム・レジスタを割り付ける方法。

(14) 更に、前記第1の物理リネーム・レジスタが前記第1の結果を記憶するために利用可能であるか否かを判断するステップと、前記第1の物理リネーム・レジスタが前記第1の結果を記憶するために利用可能であるときに、前記第1の命令を実行するステップと、前記第1の命令を実行した前記第1の結果を前記第1の物理リネーム・レジスタに記憶するステップとを含む請求項13記載の方法。

(15) 更に、前記第1の命令を完了させるステップと、前記第1の物理リネーム・レジスタから前記第1のアーキテクチャ・レジスタに前記第1の結果を書き込むステップとを含む(14)記載の方法。

(16) 更に、第2の命令を受け取るステップであって、この第2の命令を実行した第2の結果は第2のアーキテクチャ・レジスタに書き込まれるべきものであるステップと、前記第2のアーキテクチャ・レジスタに対応するように第2の仮想リネーム・バッファを割り付けるステップであって、この第2の仮想リネーム・バッファは前記第1の物理リネーム・レジスタに対する第2のポインタを含むステップとを含む(13)記載の方法。

(17) 更に、前記第1の物理リネーム・レジスタが前記第1のポインタにより指示された第1のアドレスを有するか否かを判断するステップと、前記第1の物理リネーム・レジスタが前記第1のポインタにより指示された前記第1のアドレスを有するときは、前記第1の命令を実行するステップと、前記第1の命令を実行した前記第1の結果を前記第1の物理リネーム・レジスタに記憶するステップと、前記第1の命令を完了させるステップと、前記第1の物理リネーム・レジスタから前記第1のアーキテクチャ・レジスタに前記第1の結果を書き込みステップと、前記第1のアドレスを前記第2のポインタにより指示された第2のアドレスに変換するステップと、前記第2の命令を実行するステップと、前記第1の物理リネーム・レジスタに前記第2の命令を実行した前記第2の結果を記憶するステップと、前記第2の命令の完了させるステップと、前記第2の物理リネーム・レジスタから前記第2のアーキテクチャ・レジスタに書き込むステップとを含む(16)記載の方法。

(18) 前記第1のアーキテクチャ・レジスタは、プロセッサにおける実行装置に関連したK個のアーキテクチャ・レジスタのうちの一つであり、かつ前記第1の物理リネーム・レジスタは前記実行装置に関連されたN個の物理リネーム・レジスタのうちの一つであり、かつ前記第1の仮想リネーム・バッファは前記実行装置に関連したM個の仮想リネーム・バッファのうちの一つであり、かつ $N < M$ である(13)記載の方法。

(19) 前記完了するステップは、(1)前記第1の命令、及び前記第1の命令に先行する全ての命令が実行さ

れたか、(2)割り込みが発生したか、及び(3)誤った分岐をしたかについて判断するステップを含む(15)記載の方法。

(20) プロセッサにおいて、実行装置と、前記実行装置に関連されたK個のアーキテクチャ・レジスタと、前記実行装置に関連されたN個の物理リネーム・レジスタと、M個の仮想リネーム・バッファ(ただし、 $N < M$ )と、第1の命令を受け取るディスパッチ装置であって、前記第1の命令を実行した第1の結果は前記K個のアーキテクチャ・レジスタのうちの第1のものに書き込まれるべきものである前記ディスパッチ装置と、前記K個のアーキテクチャ・レジスタのうちの前記第1のものに対応するように前記M個の仮想リネーム・バッファのうちの第1のものを割り付けるバッファ・ポインタ割り付けテーブルであって、前記M個の仮想リネーム・バッファのうちの前記第1のものは前記N個の物理リネーム・レジスタのうちの第1のものに対する第1のポインタを含む前記バッファ・ポインタ割り付けテーブルと、前記N個の物理リネーム・レジスタの前記第1のものが前記第1のポインタに対応する第1のアドレスを含むか否かを判断する回路とを含み、前記実行装置は、前記N個の物理リネーム・レジスタのうちの前記第1のものが前記第1のポインタに対応する前記第1のアドレスを含むときに、前記第1の命令を実行するものであって、更に、前記第1の命令を実行した前記第1の結果を前記N個の物理リネーム・レジスタのうちの前記第1のものに記憶する回路と、前記第1の命令を完了させる回路と、前記N個の物理リネーム・レジスタの前記第1のものから前記K個のアーキテクチャ・レジスタのうちの前記第1のものに前記第1の結果を書き込み回路とを含み、前記ディスパッチ装置は第2の命令を受け取り、前記第2の命令を実行した第2の結果は前記K個のアーキテクチャ・レジスタのうちの第2のものに書き込まれるべきものであって、前記バッファ・ポインタ割り付けテーブルは、前記K個のアーキテクチャ・レジスタのうちの前記第2のものに対応するように、前記M個の仮想リネーム・バッファのうちの第2のものを割り付け、前記M個の仮想リネーム・バッファのうちの前記第2のものが前記N個の物理リネーム・レジスタのうちの前記第1のものに対する第2のポインタを含むものであって、更に、前記第1のアドレスを前記第2のポインタにより指示された第2のアドレスに変換する回路を含み、前記実行装置は前記第2の命令を実行するものであって、更に、前記第2の命令を実行した前記第2の結果を前記N個の物理リネーム・レジスタのうちの前記第1のものに記憶する回路を含むプロセッサ。

【図面の簡単な説明】

【図1】本発明により構築されたプロセッサを示すブロック形式の図である。

【図2】本発明の一実施例を示すブロック形式

17

の図である。

【図3】プロセッサ内を通る命令の種々の段階を示すのフローチャートである。

【図4】図3に示すディスパッチ・ステップを更に詳細に示すフローチャートである。

【図5】図3に示す実行ステップを更に詳細に示すフローチャートである。

【図6】図3に示す完了ステップを更に詳細に示すフローチャートである。

【図7】本発明により構築されたデータ処理システムを示す図である。

【符号の説明】

21 BPAT (バッファ・ポインタ・アサインメント・テーブル)

\*

18

\* 22 NBPM(次バッファ・ポインタ管理) 回路

23 バッファ・ポインタ・レジスタ

24 仮想リネーム・バッファ

32 物理リネーム・レジスタ

100 プロセッサ

105 BIU (バス・インターフェイス装置)

108 命令ユニット

109 逐次フェッチャー

111 命令キュー

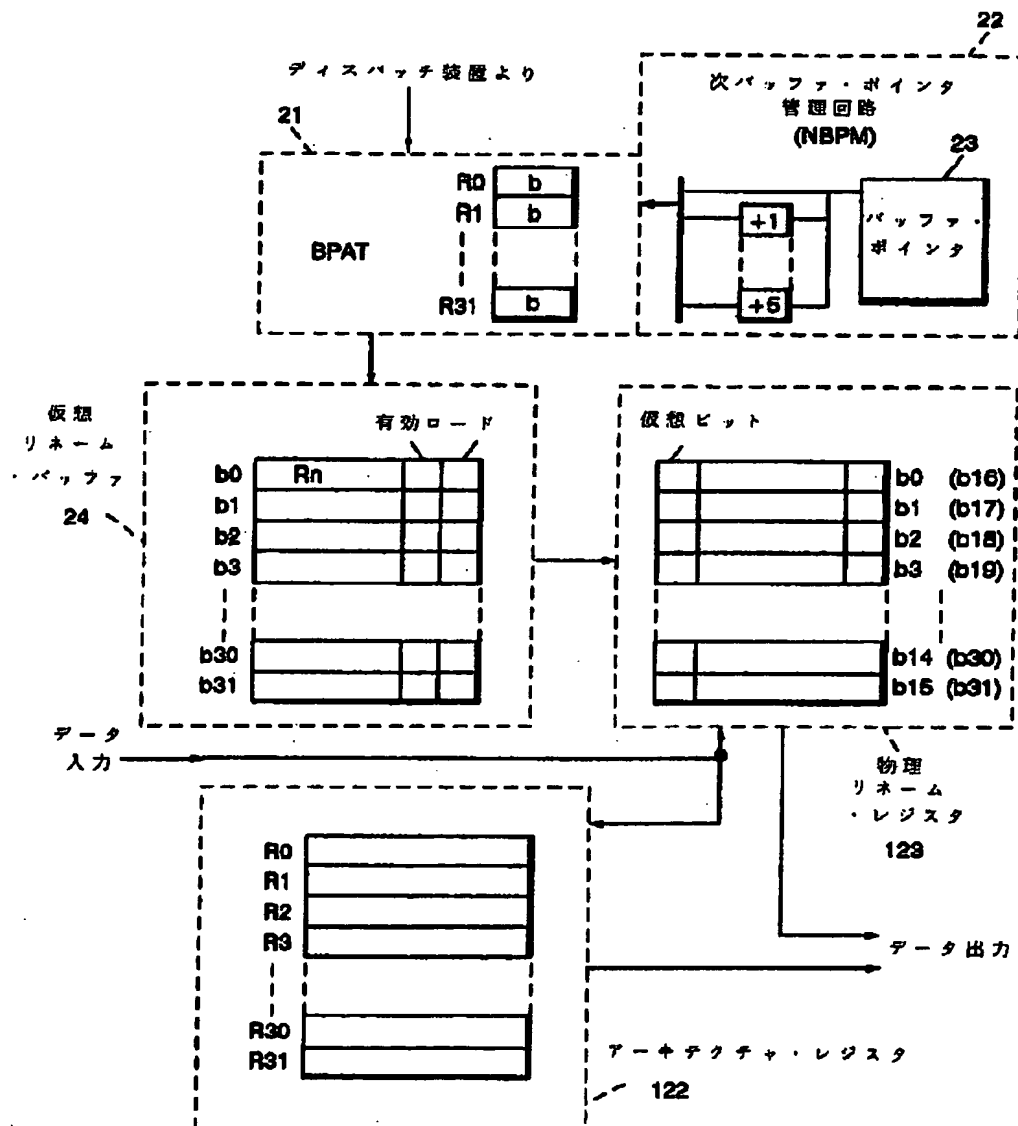
112 ディスパッチ・ユニット

117 完了ユニット

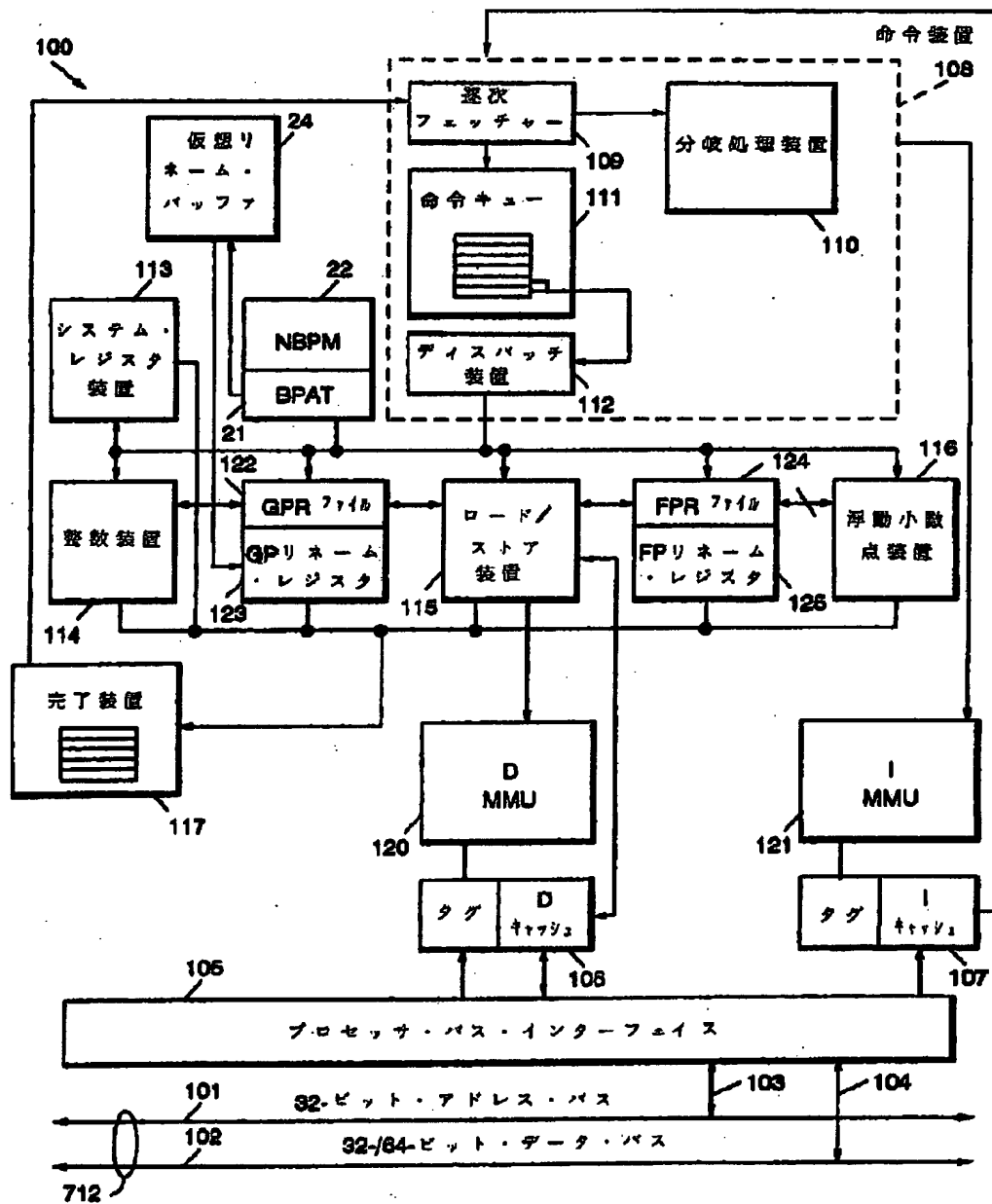
122 アーキテクチャ・レジスタ

123 物理リネーム・レジスタ

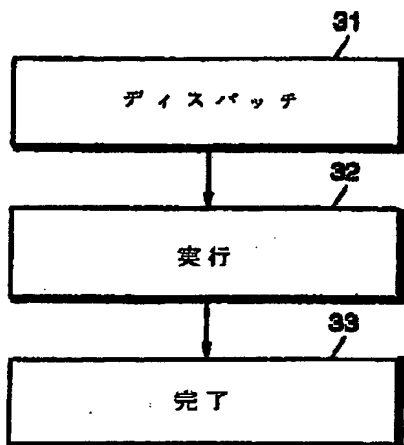
【図2】



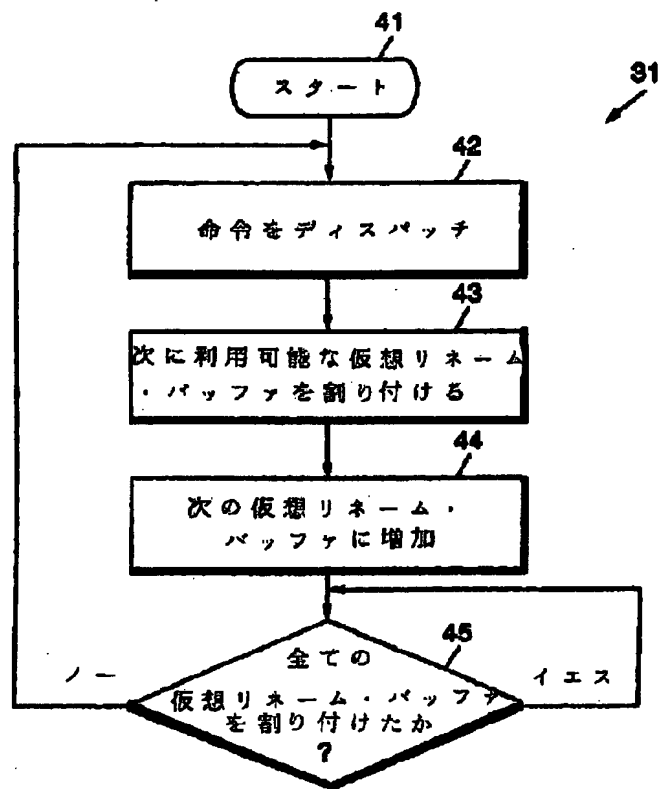
【図1】



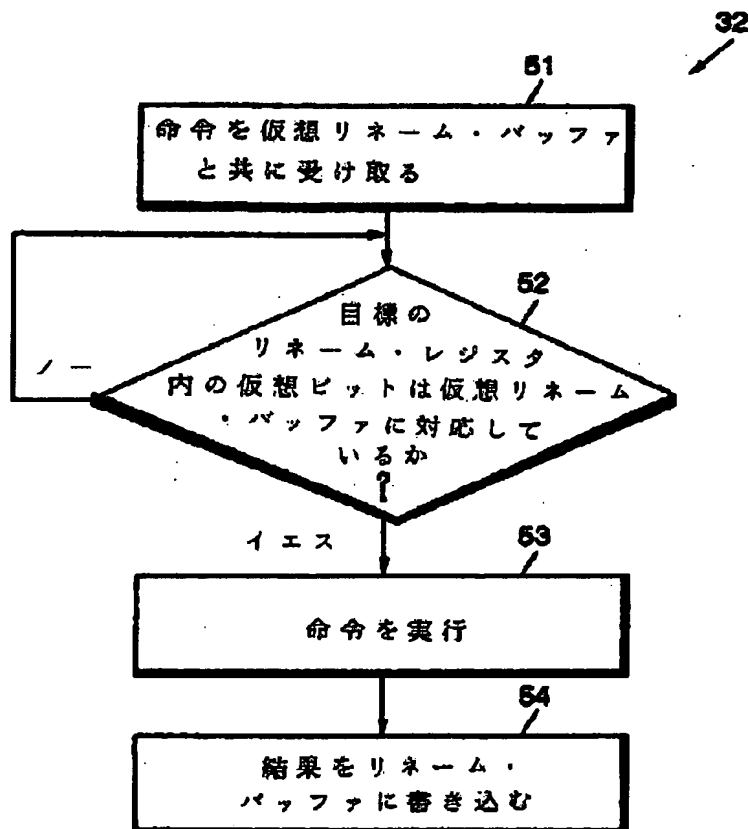
【図3】



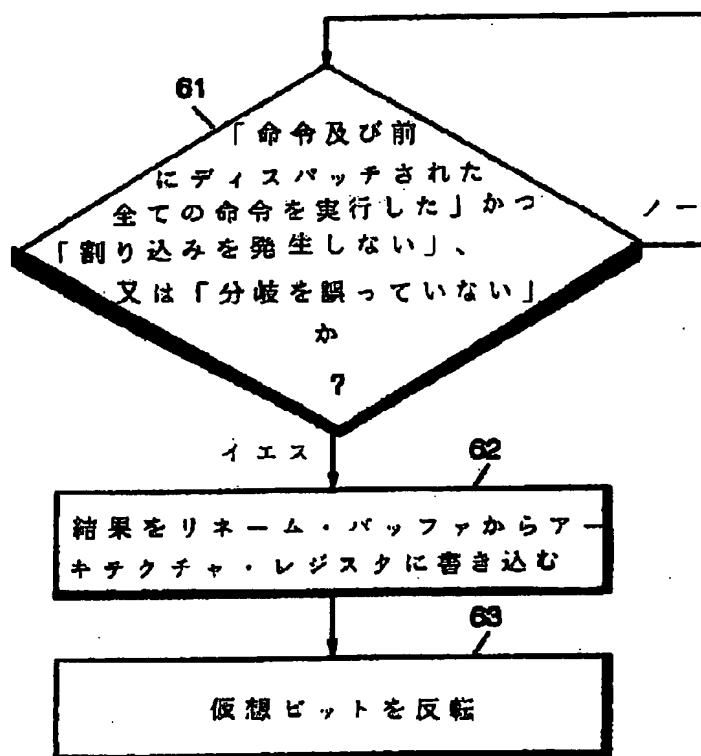
【図4】



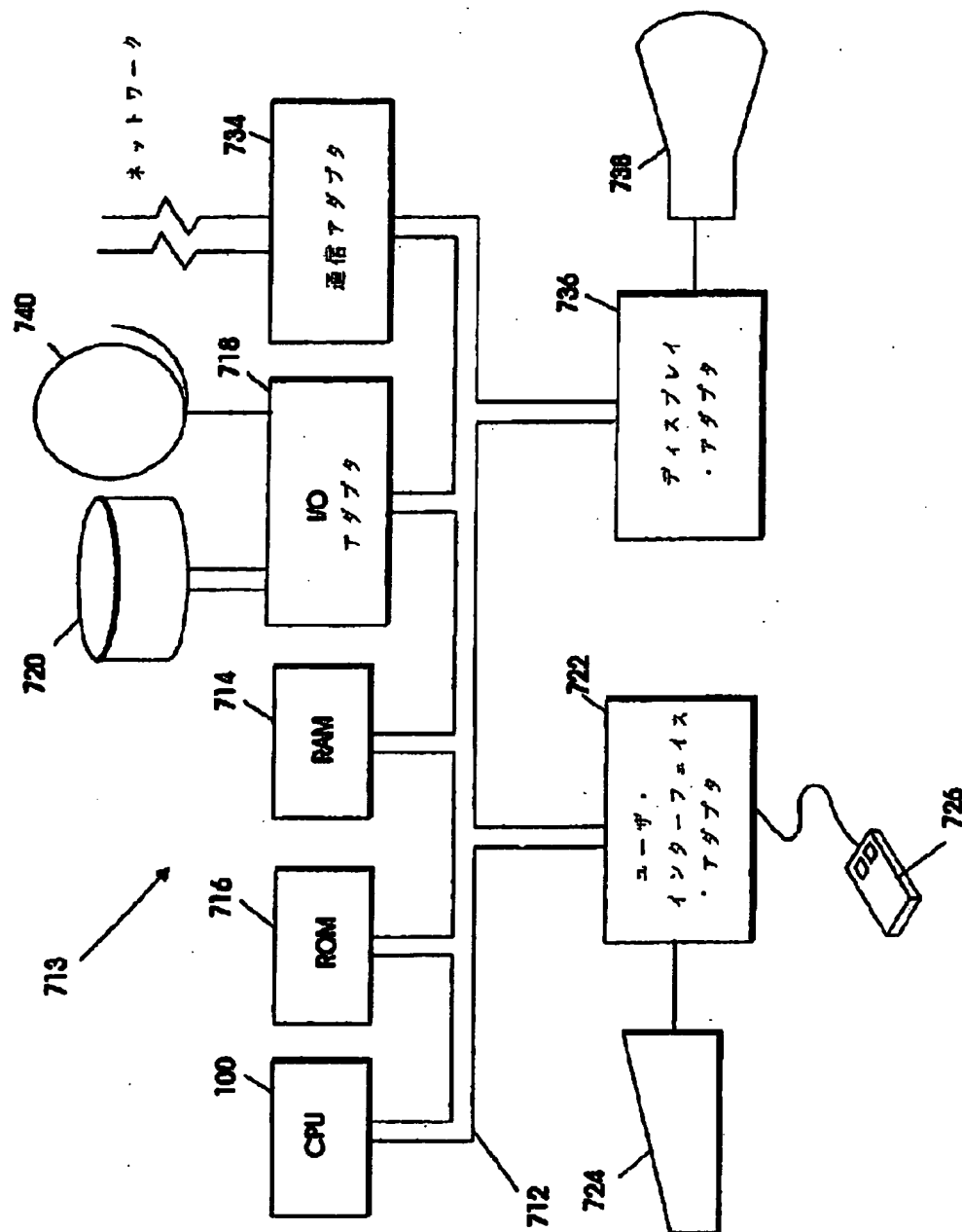
【図5】



【図6】



【図7】



フロントページの続き

(72)発明者 ハン・キュー・リー  
アメリカ合衆国78717、テキサス州、オー  
スチン、ドーマン・ドライブ 16310

(72)発明者 ダン・キュー・ングイエン  
アメリカ合衆国78728、テキサス州、オー  
スチン、ウェルドン・レーン 14308